

# Basic Python for Astronomers

## Exercises

Pim Schellart

April 18, 2013

1. The *phase* of a variable star with period  $P$  is the decimal part of  $t/P$  where  $t$  is the elapsed time since a given date. Calculate the phase when  $t = 3456.789$  days and the period is  $P = 3.1416$  days.
2. Write a script that takes three command line arguments, the first a temperature as a number, the second an input unit (C, K, F) and the third an output unit (C, K, F). Your script should then convert the input from (Celsius, Kelvin or Fahrenheit) to the specified output as in.

```
./tconvert.py 69 F C
20.6
```

3. Write a script that given three command line input numbers calculates the roots of the quadratic equation  $ax^2 + bx + c = 0$ . Make sure it works for all possible cases.
4. Spectra of stars are classified according to the letters O,B,A,F,G,K, and M. These correspond to the following temperature ranges (in degrees K):

O: 30000 - 60000	G: 5000 - 6000
B: 10000 - 30000	K: 3500 - 5000
A: 7500 - 10000	M: 2000 - 3500
F: 6000 - 7500	

Write a program which takes the temperature as a command line argument and prints out the spectral class. Print a suitable message if the temperature is out of range.

5. In 1897, the Indiana State House of Representatives passed a bill( House Bill 246) supposedly setting the value of  $\pi$  to 3. The bill was defeated in the Senate, despite support from the State Superintendent of Education. It was written by Edward J. Goodwin (who was offering his mathematical results to the State of Indiana, “free of charge”). Can you do a bit better? Write a script to calculate  $\pi$  using the series.

$$\pi^2/12 = 1 - 1/4 + 1/9 - 1/16 + 1/25 - \dots \quad (1)$$

6. Given the year, month and day of the month, the Julian day is calculated as follows:  $\text{Julian} = (36525 \cdot \text{year})/100 + (306001 \cdot (\text{month}+1))/10000 + \text{day} + 1720981$  where month is 13 for Jan, 14 for Feb, 3 for Mar, 4 for Apr etc. For Jan and Feb, the year is reduced by 1. Write a script which asks for the day, month and year and calculates the Julian day. All variables must be of integer type. What is the Julian day for 7 Jun 2008?.

7. Use the NumPy `matrix` object to solve the following linear equations.

$$x + 3y - 2z = 5 \quad (2)$$

$$3x + 5y + 6z = 7 \quad (3)$$

$$2x + 4y + 3z = 8 \quad (4)$$

8. The *bisection* method can be used for to find the root of an equation

$$f(x) = 0 \quad (5)$$

numerically, where  $f(x)$  is a continuous function of the real value  $x$  defined on an interval  $[a, b]$  when  $f(a)$  and  $f(b)$  have opposite signs. In this case  $a$  and  $b$  are said to bracket a root since, by the intermediate value theorem, the  $f$  must have at least one root in the interval  $(a, b)$ .

At each step the method divides the interval in two by computing the midpoint  $c = (a + b)/2$  of the interval and the value of the function  $f(c)$  at that point. Unless  $c$  is itself a root (which is very unlikely, but possible) there are now two possibilities: either  $f(a)$  and  $f(c)$  have opposite signs and bracket a root, or  $f(c)$  and  $f(b)$  have opposite signs and bracket a root. The method selects the subinterval that is a bracket as a new interval to be used in the next step. In this way the interval that contains a zero of  $f$  is reduced in width by 50% at each step. The process is continued until the interval is sufficiently small, or until a specified number of iterations have been performed. Explain why these two conditions are equivalent.

Explicitly, if  $f(a)$  and  $f(c)$  are opposite signs, then the method sets  $c$  as the new value for  $b$ , and if  $f(b)$  and  $f(c)$  are opposite signs then the method sets  $c$  as the new  $a$ . (If  $f(c) = 0$  then  $c$  may be taken as the solution and the process stops.) In both cases, the new  $f(a)$  and  $f(b)$  have opposite signs, so the method is applicable to this smaller interval.

Now plot the polynomial function

$$g(x) = 3x^3 - \frac{23}{2}x^2 + 3x \quad (6)$$

and the line  $h(x) = -4$ . Write a function `my_bisect` that takes as arguments; a function `f`, the start of the interval `a`, the end of the interval `b`, and a number of steps `nsteps`. The function then uses the above algorithm to find the root of the function `f` on the interval  $[a, b]$ . Use this function to calculate every  $x$  where  $g(x) = h(x)$ .

Now use the function `bisect` from the `scipy.optimize` package to check your answers.

9. Here we will play a bit with random numbers. You can generate an array of random numbers with the `rand` function from the `numpy.random` package. Use it to get an array containing 10000 random numbers between  $-3$  and  $5$ . Now plot a 50-bin histogram of this distribution with the `matplotlib.hist` function. Be sure to label the axes properly. What does this distribution look like? What if we now take the mean of 100 random numbers in the range  $[0, 1)$  and we do this a thousand times. Try to guess what the distribution of the mean values will look like? Test this by plotting a histogram of the distribution of the mean values. This is an example of the central limit theorem that you will learn more about next week.